

MIP: The ITS-EPFL Medical Image Processing Library

M. Bach Cuadra, T. Butz, O. Cuisenaire, E. Solanas, J.-P. Thiran

Signal Processing Institute (ITS)

Swiss Federal Institute of Technology (EPFL)

CH-1015 Lausanne, Switzerland

Email: {Meritxell.Bach,Torsten.Butz,Olivier.Cuisenaire,Eduardo.Solanas,JP.Thiran}@epfl.ch

<http://ltswww.epfl.ch/~brain>

ITS Technical Report 02.08

Abstract— This document presents a brief description of the Medical Image Processing (MIP) library developed by the Brain Group of the Signal Processing Institute (ITS) at Swiss Federal Institute of Technology (EPFL). This library is a result of more than three years of research by Meritxell Bach Cuadra, Torsten Butz, Eduardo Solanas and Olivier Cuisenaire, carried out under the supervision of Jean-Philippe Thiran. You can find more detailed information about the library at our web page located at <http://ltswww.epfl.ch/~brain> in the Manuals section.

I. INTRODUCTION

The MIP library was created to optimize our programming efforts by maximizing mutual compatibility between different utilities for medical image processing and by minimizing overlap between different projects. The initial idea consisted of just sharing C++ objects, mainly for data structures commonly employed within medical imaging algorithms. However, the MIP library is evolving along with our research and by now, it has become a very flexible and powerful tool for developing image processing algorithms. The implemented structure hardly requires a priori knowledge of the specific implementations and has therefore the potential to be of help for a large group of people, even outside the medical imaging community.

The structural simplicity is an important characteristic of our library and allows easy and efficient parallel implementation of image processing algorithms. In particular *Message Passing Interface* (MPI)¹ implementations are heavily facilitated as we provide some MPI-utilities.

The MIP library contains many modules with different functionalities. Concretely this means that we generate a dynamic library for every functional group of classes (see next section for more details). The time spent to learn the usage of the MIP library is more than compensated

by the gain of time of having a large class of important functionalities already implemented.

II. GENERAL DESCRIPTION

As mentioned, the MIP library is subdivided in dynamic libraries generated separately for every functional group of classes. The initial division contained only three different groups: one for the classes corresponding to *data structures*, one for general *processing filters* which contains the processing classes such as a Gaussian filtering or Gradient filter and finally the *statistical utilities* containing statistical optimization objectives, such as mutual information, and other statistical utilities.

Since the beginning, this basic structure has evolved considerably. We still have these three groups, but many more modules have been added. In the following we present a brief description of each functional module. Please refer to the Manuals section of our web-page for a more detailed description or additional information about a specific class.

A. Data sets

This group of classes contains data representations for different widely used geometries. We have the initial ancestor *DataSet* from which sub-classes are successively derived to differentiate different dimensions and geometries. Then we attach different possible data objects to the defined geometries, so far scalars or vectors, resulting in the final data representations.

B. Data objects

This group contains data objects, such as matrices or vectors. Potentially these classes provide the data objects to be attached to the geometries of the data sets of the previous paragraph. This migration has not been completed yet.

¹<http://www-unix.mcs.anl.gov/mpi/>

C. Processing

This is the biggest module of our library. All the objects are derived from the ancestor *Filter*. The module contains filters such as a Gaussian filter or an anisotropic filter but also other objects to calculate non-rigid and affine transformations or to use B-spline interpolation. Also a filter to calculate the distance map of a binary object and a resampling class that performs the down or up-sampling of data sets are contained in this group.

D. Statistics

The objects contained in this module all refer to statistics. We distinguish first the structures corresponding to a sum of probability densities and then the filters that calculate the histogram and the joint histogram. Finally we have some modules that calculate different measures such as the first and second statistical moments, the entropy or the mutual information.

E. Optimization

Optimization can be a computationally very expensive task. We therefore provide some specialized parallel optimization algorithms, in particular steepest gradient and genetic optimization.

F. MPI utilities

For further easy parallelization, we provide some very useful MPI utilities.

G. Neural Networks

This module allows very easy topological construction of neural networks. So far no optimization is associated, so either we inherit to more specialized neural networks or we simply use the optimization algorithms of other modules.

H. Numerical Recipes in C

Our library contains several functionalities of the Numerical Recipes in C library².

I. Utilities

This module contains the file to include the needed header files, some macros and global definitions. It contains also a module for swapping bytes depending if the system uses the little endian or big endian reference and overloaded functions to determine the type (float, integer, etc.) of any C data variable.

J. Reader and writer

The input-output module differentiates reader and writer structures. We have for each different data set object a reader and a writer structure that needs a file name and the structure to read or write. These modules considers writing or reading several file formats such as .h33, .ima, .vol, .pgm, or matlab files.

²<http://www.nr.com/>

K. Main programs

The group of main programs contains for the moment the kNN classification, a bias corrector module and the 3D non-rigid deformation based on optical flow.

L. Windows

Then win32 directory contains the appropriate Makefiles to allow compiling MIP under windows using cygwin³ and gcc. It creates the mipdll.dll dynamic library and explains how to compile your own programs and link them with mipdll.

III. RESEARCH DONE USING MIP LIBRARY

Here we present the list of published work that used the MIP library. It contains many different key research areas of medical imaging, or image processing in general.

1. Torsten Butz, Olivier Cuisenaire and Jean-Philippe Thiran. **Multi-modal Medical Image Registration: From Information Theory to Optimization Objective**. DSP 2002, 407-414. Santorini, Greece, 2002.
2. Torsten Butz and Jean-Philippe Thiran. **Feature Space Mutual Information in Video-Speech Sequences**. Accepted for ICME 2002. Lausanne, Switzerland, 2002.
3. Torsten Butz and Jean-Philippe Thiran. **Feature-space mutual information for multi-modal signal processing, with application to medical image registration**. Accept for EUSIPCO 2002. Toulouse, France, 2002.
4. P. Hagmann, S. Clarke, J.-P. Thiran, P. Maeder, R. Meuli. **Variation in fiber spreading from the human corpus callosum: a statistical study on DT-MRI data**. OHBM 2002, Sendai.
5. M. Bach Cuadra, B. Platel, E. Solanas, T. Butz, J.-P. Thiran. **Validation of Tissue Modelization and Classification techniques in T1-weighted MR Brain Images**. Accepted at MICCAI 2002. Tokyo, Japan, 2002.
6. M. Bach Cuadra, J. Gomez, P. Hagmann, C. Pollo, J.-G. Villemure, B. M. Dawant, J.-Ph. Thiran. **Atlas-based Segmentation of Pathological Brains using a Model of Tumor Growth**. Accepted at MICCAI 2002. Tokyo, Japan, 2002.
7. Torsten Butz and Jean-Philippe Thiran. **Shot Boundary Detection with Mutual Information**. ICIP 2001, 422-425. Thessaloniki, Greece, 2001.
8. Eduard Solanas, Valerie Duay, Olivier Cuisenaire and Jean-Philippe Thiran. **Relative Anatomical Location for Statistical Non-parametric Brain Tissue Classification in MR Images**. ICIP 2001, 885-888. Thessaloniki, Greece, 2001.
9. Meritxell Bach Cuadra, Olivier Cuisenaire, Reto Meuli and Jean-Phillippe Thiran. **Automatic Segmentation of Internal Structures of the Brain in MR Images Using a Tandem of Affine and Non-Rigid Registration of an Anatomical Brain Atlas**. ICIP 2001, 1083-1086. Thessaloniki, Greece, 2001.

³<http://www.cygwin.com>

10. Torsten Butz and Jean-Philippe Thiran. **Affine Registration with Feature Space Mutual Information**. MICCAI 2001, 549-556. Utrecht, NL, 2001.
11. Eduard Solanas and Jean-Philippe Thiran. **Exploiting Voxel Correlation for Automated MRI Bias Field Correction by Conditional Entropy Minimization**. MICCAI 2001, 1220-1221. Utrecht, NL, 2001.

IV. CONCLUSION

We presented the Medical Image Processing library which was created and developed at the Signal Processing Institute of EPFL. This library is of course still under construction since it's evolving every day. Anyway, we will present soon in our web-page a free version of the MIP library to be freely distributed for academic research.

V. ACKNOWLEDGMENTS

We would like to thank all the students that have collaborated in the development of the library.